PATENT **Attorney Docket No. SVL920010003US1 (13296/2)**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/990,802	Confirmation No.:	3720
Applicant(s)	:	Eitan FARCHI et al.		
Filed	:	November 13, 2001		
For	:	METHOD AND APPARATUS FOR COLLECTING PERSISTENT COVERAGE DATA ACROSS SOFTWARE VERSIONS		
Art Unit	:	2193		
Examiner	:	Jason D. MITCHELL		
Customer No.	:	61023		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR RECONSIDERATION

Sir:

Applicants respectfully request reconsideration of the Board Decision dated February 16, 2007, affirming the Examiner's rejections of the claims in the subject application.[1]

In their Brief on Appeal, and in their Reply Brief, Appellants advanced a number of arguments for the patentability of the claims of the present application. In affirming the Examiner's rejections, the Board obviously sided with the Examiner and decided against Appellants. However, Appellants respectfully submit that the Board's Decision does not give sufficient guidance to Appellants as to why the many arguments they advanced were somehow insufficient, improper, or incorrect. Indeed, the Board Decision appears not to have addressed a

[1] While this paper is dated outside of the nominal two-month response period, Applicants have filed a Petition to Revive the subject application, based on their non-receipt of the Board Decision.

number of arguments that Appellants made, arguments that Appellants believe should have led to a different decision from the Board. Accordingly, Appellants respectfully request reconsideration of the Board Decision.

In addition, assuming that the Board decides to maintain the present affirmance of the Examiner's rejection, Appellants respectfully submit that the Board Decision is not sufficiently detailed to enable Appellants meaningfully to determine how best to proceed, should Appellants decide to appeal the Board Decision. Accordingly, and as an alternative ground for reconsideration, Appellants respectfully request that the Board clarify, where possible, why Appellants' arguments do not suffice to overcome the points the Examiner made.

A listing and discussion of the relevant points follow.

1) In the Reply Brief, Appellants made two separate arguments (Reply Brief, page 2) as to why the Examiner's position that Chen taught the persistence of code coverage tasks across software versions was incorrect. In both of these instances, Appellants pointed out that the code would only exist in one version until the next version was created. Once the next version was created, the code would no longer exist. If the code ceases to exist in a subsequent version when it existed in the immediately preceding version, clearly the code does not persist across software versions. "Across" implies a bridge or somehow existing on both sides of something. If code ceases to exist once a new version is created, clearly that codes exists on one side (the earlier version), but not on the other side (the later version). Therefore, the code does not persist as claimed.

While the Board said the Examiner found properly that the code persisted across code versions, the Board did not address Appellants' detailed argument in this regard, and in particular

did not appear to address specifically Appellants' argument that "across" means "on both sides of," while Chen clearly teaches something less than that.

2) In the Appeal Brief, on pages 11 and 12, Appellants differentiated in detail between Chen's "basic code entities," which are functions or non-functions, and the claimed "code coverage tasks," which are defined on pages 12 and 13 of the present application as something different from functions and non-functions. Specifically, in the course of describing Figure 2 and steps 202 and 204, Appellants define the term "code coverage task" as follows:

> A code coverage task is a basic block of code for which an execution of a test returns a true value if the testing requirement of the task is fulfilled and a false value if the testing requirement of the task is not fulfilled. A basic block is a set of consecutive statements with a single entry point (i.e. the first statement) and a single exit point (i.e. the last statement). Control statements such as the "if" statement are considered as a separate block to ease the detection of source code changes that affect the associated blocks (i.e. basic blocks which follow the control statement). Source code changes will be discussed in more detail later. Those of ordinary skill in the art will recognize that there are other alternative ways to divide a program source code into coverage tasks. For example, coverage tasks could be at module level, block level or statement level and could be identified manually rather than automatically and could be based on the user's needs.
>
> (emphasis added)

Chen's description of "basic code entities" as functions or non-functions in no way equates those entities to Appellants' claimed "code coverage tasks".

In the Decision on Appeal, the Board agreed with the Examiner that Chen's "basic code entities" were the same as Appellants' claimed "code coverage tasks," but did not say why Appellants were somehow incorrect in their assertion. Clearly, "code coverage tasks" is a term which Appellants have coined in their specification, as they are entitled to do as their own lexicographers. Appellants clearly pointed out what that definition was from the specification, and how it differentiated from Chen's basic code entities. The Examiner never asserted that the term "code coverage tasks" was a term of art which was somehow synonymous with Chen's basic code entities, nor that the ordinarily skilled artisan would view the terms as synonymous. Clearly, the terms comprehended different things, as Appellants pointed out in detail in their Appeal Brief.

In siding with the Examiner, the Board never made a finding, nor gave any indication why Chen's "basic code entities" were somehow the same as Appellants' claimed "code coverage tasks," but instead merely agreed with the Examiner. In this regard, Appellants note that the Examiner's arguments never really addressed Appellants' arguments about why a code coverage task was not a basic code entity.

3) In the paragraph bridging pages 12 and 13 of their Appeal Brief, Appellants identified numerous claim elements which could not read on Chen, given the difference between code coverage tasks and Chen's basic code entities. While the Board chose to focus only on the aspect of the invention related to the persistence of code coverage tasks, Appellants' discussion of code coverage tasks took into account the dividing, inserting, and creating steps in

independent claim 1 and independent claim 8 and independent claim 15. The Board's Decision addressed none of this. Again, the Board's Decision focused only on the aspect of persistence.

4) Regarding the rejection based on Winder, the Board characterized Appellants' argument as being no more than a statement that Winder did not supply the deficiencies of Chen. However, on pages 13 and 14 of their Appeal Brief, Appellants discussed why it would not have been obvious to modify Chen in view of Winder. Appellants pointed out, for example, that Winder has nothing to do with code coverage tasks or basic code entities. Appellants went on to state that Winder served no purpose with respect to Chen because Winder contributed, at most, naming conventions which would be part of writing source code in any event. The Board's Decision says nothing about these arguments regarding Winder.

Pursuant to the foregoing, Appellants respectfully request the Board reconsider its decision, and either reverse the Examiner's rejection, or provide clarification as to why Appellants' arguments are incorrect or insufficient.

Respectfully submitted,
KENYON & KENYON LLP

Dated: June 20, 2007

By: /Frank L. Bernstein/
Frank L. Bernstein
Reg. No. 31,484

Customer No. **61023**

KENYON & KENYON LLP
333 West San Carlos St., Suite 600
San Jose, CA 95110
Telephone: (408) 975-7500
Facsimile: (408) 975-7501